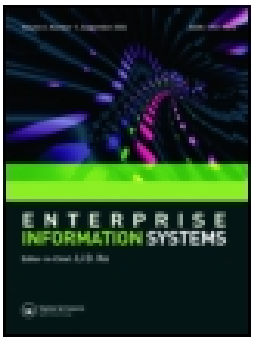


Enterprise Information Systems

ISSN: 1751-7575 (Print) 1751-7583 (Online) Journal homepage: http://www.tandfonline.com/loi/teis20

Scheduling multimedia services in cloud computing environment

Yunchang Liu, Chunlin Li, Youlong Luo, Yanling Shao & Jing Zhang

ENTERPRISE INFORMATION SYSTEMS, 2017
http://dx.doi.org/10.1080/17517575.2017.1309686

Taylor & Francis
Taylor & Francis Group

Check for updates

Scheduling multimedia services in cloud computing environment

Yunchang Liu[a], Chunlin Li[a,b], Youlong Luo[a], Yanling Shao[a] and Jing Zhang[a]

[a]Department of Computer Science, Wuhan University of Technology, Wuhan, P.R. China; [b]Key Laboratory of Spatial Data Mining & Information Sharing of Ministry of Education, Fuzhou University, Fuzhou, P.R. China

ABSTRACT

Currently, security is a critical factor for multimedia services running in the cloud computing environment. As an effective mechanism, trust can improve security level and mitigate attacks within cloud computing environments. Unfortunately, existing scheduling strategy for multimedia service in the cloud computing environment do not integrate trust mechanism when making scheduling decisions. In this paper, we propose a scheduling scheme for multimedia services in multi clouds. At first, a novel scheduling architecture is presented. Then, We build a trust model including both subjective trust and objective trust to evaluate the trust degree of multimedia service providers. By employing Bayesian theory, the subjective trust degree between multimedia service providers and users is obtained. According to the attributes of QoS, the objective trust degree of multimedia service providers is calculated. Finally, a scheduling algorithm integrating trust of entities is proposed by considering the deadline, cost and trust requirements of multimedia services. The scheduling algorithm heuristically hunts for reasonable resource allocations and satisfies the requirement of trust and meets deadlines for the multimedia services. Detailed simulated experiments demonstrate the effectiveness and feasibility of the proposed trust scheduling scheme.

ARTICLE HISTORY
Received 2 September 2015
Accepted 6 February 2017

KEYWORDS
Trust mechanism; cloud computing; multimedia services; QoS

1. Introduction

To meet the requirements of users, many of multimedia service providers (MSPs) provision various multimedia services such as, WebTV, VoD, mobile instant message, voice and video services delivered over Internet (Tan and Su 2011). Multimedia services demand power computational capacity and large storage space to process rich multimedia content. Recently, cloud computing has broad prospect to provide desired resources and provisioning quality of service (QoS) for multimedia services due to its features of on-demand supply and cost advantage (Tan and Su 2011; Zhu et al. 2011). Many of multimedia services in cloud computing environment spring up, such as video surveillance (Ren and Van Der Schaar 2013), cloud-based healthcare (Shini, Thomas, and Chithraranjan 2012), real-time access of medical images and video (An et al. 2014; Yang et al. 2015), cloud based photo and video sharing and so on.

Currently, security is a critical factor for multimedia services running in cloud computing environment (Lu, Varna, and Wu 2010; Wang et al. 2014b; Zhu and Lu 2009). Due to inherent distributed deployment of cloud computing environment, it is difficult to ensure each user can recognize a huge number of MSPs and select appropriate and trustworthy MSPs for trade.

CONTACT Chunlin Li chunlin74@aliyun.com Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P.R. China

Furthermore, uncertainty and unreliability of multimedia cloud pose threats to the operation of multimedia services. For example, multimedia cloud services sometimes may be offline unexpectedly due to power outage or other reasons. And the abuses of service resources by malicious users will greatly deplete MSP's resources and decrease its service capabilities (Zhou et al. 2011). On the other hand, some of selfish MSPs focus on augmenting their benefit and do not worry the benefit of users. In the context, it has contributed to the lack of confidence in the information discovered by the MSPs (Fard, Prodan, and Fahringer 2013). For instance, the MSPs might overstate the capacity of their resources in order to achieve more profits.

As an effective mechanism, trust has recently been suggested to enhance security and alleviate attacks in cloud computing environments (Dong and Dai 2008). Executing the jobs on the trusty MSPs can guarantee the security of sensitive data, reduce the services completion time and drop the ratio of failure execution. Unfortunately, most existing scheduling algorithms for multimedia services in cloud computing environment are merely devised to guarantee QoS demands while possibly ignoring the trust.

In this paper, we propose a novel scheduling and provisioning architecture for multimedia services in cloud computing environment. Within the proposed architecture, we build a trust model including both subjective trust and objective trust to evaluate the trust degree of a MSP. By employing Bayesian theory, the subjective trust degree between multimedia service providers and users is obtained. According to the attributes of QoS, the objective trust degree of a MSP is calculated. Considering QoS demands in terms of cost, complete time and the requirement of trust for multimedia services in cloud environment, we propose Trust Multimedia Scheduling Strategy (TMSS), a novel trust mechanism that counters the sensitive security of multimedia services. TMSS can heuristically find trusted resource allocations and achieve minimum completion time while meeting the constraints of cost and trust.

The contributions of this paper include the following aspects:

(1) A novel scheduling and provisioning architecture for multimedia services running in cloud environment is designed.
(2) TMSS is proposed for security-sensitive multimedia services with independence tasks in such cloud computing environment.
(3) Detailed simulation experiments illustrate the validity and practicable of the proposed scheme.

The rest of the paper is organized as follows: Section 2 summarizes the related work. In Section 3, we describe the scheduling architecture for multimedia services in cloud computing environment. In Section 4, we describe the formulation and solution of multimedia service scheduling and provisioning model, and present a novel trust model including subjective and objective trust that can compute the trust degree of MSPs and users. The proposed scheduling algorithm called TMSS is discussed in Section 5. Simulated experiment results of the proposed scheme are analyzed in Section 6. Section 7 makes the conclusion and illustrates the future work.

2. Related works

In this section, we briefly summarize the related work about cost optimal scheduling, load balancing scheduling, resource failure and trust mechanism for multimedia services in cloud computing environment.

A number of scheduling algorithms for multimedia service have been developed to satisfy industrial requirements. Zhou et al.(2011) proposed a security-critical heuristic strategy to schedule multimedia applications on heterogeneous networks. Security requirements for multimedia applications is integrated into a real-time scheduling algorithm. Jyh-Cheng et al. (1998) investigated how to schedule of multimedia services in a low-power wireless ATM networks. The proposed algorithm is a priority round robin with dynamic reservation update scheduling. Dong and Dai (2008) studied how to schedule non-real-time or real-time multimedia services in IEEE 802.16 e OFDM system. Considering the performances of efficiency, fairness and complexity, they proposed a new scheduling algorithm improving the efficiency of air interface resource at the expense of the short-time unfairness, but ensures the long-time fairness. Wu et al. (2010) built a model to realize the e-customer relationship building in buying multimedia product, suggest watermarks can help relationship building and retain e-customer to obtain multimedia products from legal channel.

Cloud resource providers provide on-demand resource provisioning with a pay-per-use model. Due to stronger resource demands of multimedia services, there is an upsurge of research interest in cost optimal for multimedia services in cloud computing environment. Li, Zhuo, and Shen (2013) presented a QoS based resource allocation method for handling multimedia tasks in the cloud. Considering completion time, cost and energy consumption, a QoS-based resource allocation method is proposed. Based on queuing model, Nan, Yifeng, and Ling (2011) proposed a novel optimized resource allocation schema for multimedia cloud. The response time minimization problem and resource cost minimization problem is formulated and solved in single service and multiple services, respectively. Zhipiao Liu et al. (Dong and Dai 2008) established a cloud service request model with SLA constraints, and then presented a cost-aware service request scheduling approach based on genetic algorithm. Wang et al. (2014c) proposed an accurate evaluation approach of Quality of Cloud Service (QoCS) in service-oriented cloud computing. The authors employed fuzzy synthetic decision to evaluate cloud service providers according to cloud users' preferences and then adopt cloud model to computing the uncertainty of cloud services based on monitored QoCS data. Wang et al. (2016) explored an alternative virtual machine placement approach to minimize energy consumption during the provision of data-intensive services with a global QoS guarantee in national cloud data centers. They used an improved particle swarm optimization algorithm to develop an optimal virtual machine placement approach involving a tradeoff between energy consumption and global QoS guarantee for data-intensive services. Song et al. (2016) presented a two-stage approach: media task management and cloud resource management for multimedia services in a cloud computing environment. The concept of allocation deadline is introduced in both of these approaches which makes the proposed solution unique from existing methods and benefits both multimedia service provider and cloud operator. For the media task management, a queuing based approach is proposed. In this approach, an online and efficient task management algorithm using Markov analysis and prediction was developed. For the resource management, an efficient heuristic algorithm was presented that can select and allocate task in an active and dynamic way through a dynamic controller. Lu et al. (2015) studied how to maximize an MSP's profit from provisioning multimedia services to geographically distributed users with a hybrid cloud. They first designed a service provisioning model to manage the resources in the hybrid cloud. Considering different time granularities for resource reservations, they leveraged the Lyapunov optimization technique to maximize the profit of MSP and proposed an online algorithm that can manage the hybrid cloud in the distributed manner. Briefly, some approaches described above assume that the resource information published by the MSPs is correct, which is doubtful to selfish MSPs.

Load balancing for multimedia services in cloud computing environment have been widely studied. Wen et al. (2011) investigated how to make each multimedia service task to obtain the required resources in the shortest time. They proposed an effective load balancing approach for cloud-based multimedia system. Its main advantage is fully considering the load of all servers and the network conditions, and thus achieving reasonable resource allocation and scheduling. Lin, Chin, and Deng (2014) took into account a more practical dynamic multi service scenario in which each server cluster only handles a specific type of multimedia task, and each client requests a different type of multimedia service at a different time. The scenario can be modeled as an integer linear programming problem. In (Lin et al. 2013), cloud computing based multi-tasking video transcoding is investigated and a parallelizing video transcoding framework is proposed based on load balancing strategy. The multi-resources transcoding process in the cloud as a NP-hard problem was formulated. A heuristic algorithm called MLFT is proposed to minimize the entire finish time. In summary, some of aforementioned scheduling algorithms do not consider the dynamic change of resources in a real cloud computing.

Due to the increased functionality and complexity of the cloud computing system, resource failures are inevitable. Wang et al. (2014a) studied the performance of a cloud-based multimedia system using retrying for fault tolerance. The response time of the cloud-based multimedia system is modeled and the probability distribution of the response time is derived considering the effect imposed by retrying tasks in case of service failures happening. Javadi, Abawajy, and Buyya (2012) proposed a scalable hybrid Cloud infrastructure as well as resource provisioning policies to assure QoS targets of the users. The proposed policies take into account the workload model and the failure correlations to redirect users' requests to the appropriate Cloud providers. Though the works consider the resource failure and provision reliable multimedia services for user, they ignore security or trust requirement in scheduling schema and decrease the efficiency of media task scheduling.

At present, trust has been regarded as a valid mechanism to promote security within networked environments such as grid, cloud computing. Based on cognitive trust model, Wang et al. (2012) evaluated the trust of entities employing Bayesian theory, and integrated the trust model into existing DLS (dynamic level scheduling) algorithm to schedule the directed acyclic graph (DAG)-structure applications. In (Qu and Buyya 2014), Qu C. and Buyya R. proposed a system evaluating trust of clouds with regard to users' fuzzy QoS requirements and services' dynamic performances to promote service selection. Employing differential equations, Xiaoyong et al. (2011) designed a security scheduling schema to evaluate the trust degree of each node in heterogeneous system. It employed task priority rank to figure out security overhead of critical tasks. A security scheduling algorithm for DAG is presented to meet security requirement of workflow applications. Li et al. (2014) considered cloud user's trust and QoS and focused on a trustful workflow scheduling schema in multi cloud environment. The proposed schema describes the trust mechanism between cloud service providers and users. The trust mechanism can assure the successful execution of critical tasks. Two-level based scheduling mode maybe easily handles with the problem of workflow scheduling. Fard, Prodan, and Fahringer (2013) presented a pricing model and a truthful strategy to schedule single tasks. In order to minimize the completion time and monetary cost, this paper developed a trust scheduling algorithm for scientific workflows. Tan et al. (2014) proposed a trust service-oriented workflow scheduling algorithm. The scheduling algorithm uses Fuzzy Model to formulate the scheduling problem. It provided various strategies to help users balance different requirements in terms of cost, deadline and trust. Compared with our research, few of work focus on the problem of trust scheduling for multimedia services in cloud computing environment.

3. Multimedia services scheduling and provisioning architecture in cloud environment

In this section, we first present the multimedia services scheduling and provisioning architecture in cloud computing environment, and then we describe the scheduling process for multimedia services.

As depicted in Figure 1, the entities in cloud computing environment consist of MSPs, users and credible intermediate service center (CISC).

MSPs provision various multimedia services to end users for profit. MSPs offering the similar multimedia services are classified as one class. MSPs can resort to other MSPs for service composition in order to satisfy users' demands. Users access and obtain multimedia services provisioned by MSPs. Users are categorized according to their social contexts, preference for sharing services, and demand of available resource. CISC is the service register center of multimedia cloud environment. MSPs must register and publish a variety of multimedia services in

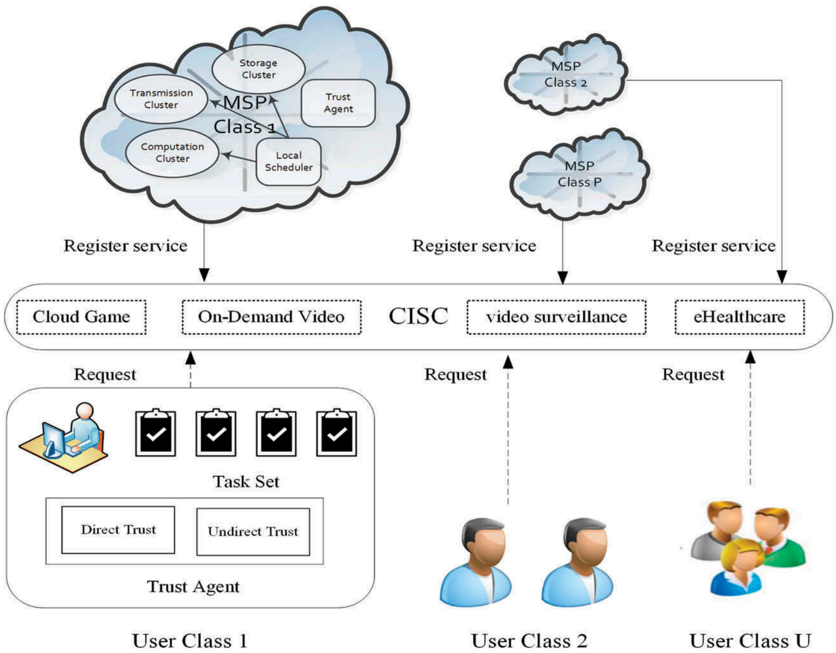


Figure 1. Multimedia services scheduling architecture in cloud computing environment.

CISC. In additional, CISC is the trustful third party intermediaries. Users resort CISC to find suitable MSPs according to the information provided by CISC in term of current users, resource and system load.

Users' trust agent evaluates the integrity of the MSPs according to the trust degree and informs user which is the most suitable one and execute the trade. Trust Agent uses Bayesian trust model, Eigen Trust (Kamvar, Schlosser, and Garcia-Molina 2003), differential equation (Xiaoyong et al. 2011) to compute subjective trust degree based on the information from direct experience or recommend reply. In additional, user can obtain the objective trust degree of a MSP from the feedback information of CISC. Finally, according to the subjective and objective trust degree of a MSP, users calculate the comprehensive trust value and determine whether they can trade with the special MSP. MSPs like users also judge the users' trust degree when processing services. MSPs can select trustworthy MSPs for service composition to meet personalized requirements.

The multimedia services scheduling process is detailed as followed.

When one user wants to trade with one MSP, he (she) submit its requests to the CISC, then CISC will help the user to select trustful MSPs. The user trade with MSPs and obtain the direct trust. The direct trust varies by MSPs and user's direct interaction, environment changing, and declines with time. The user first checks transaction history with the MSP. If the direct trust threshold is smaller than the effective number, he (she) take trust decision according to the direct trust information, else request recommendations from the same type of other users and feedback trust from CISC.

If a MSP processes service requests submitted by one user, he uses its trust agent to determine whether the user is trustful. If the requester is not included the untruthful list, the trust agent suggest the MSP to trade with the user. The trust relation among MSPs is same to the one that exists between users and MSPs.

Figure 2 describes the multimedia services scheduling process in cloud computing environment.

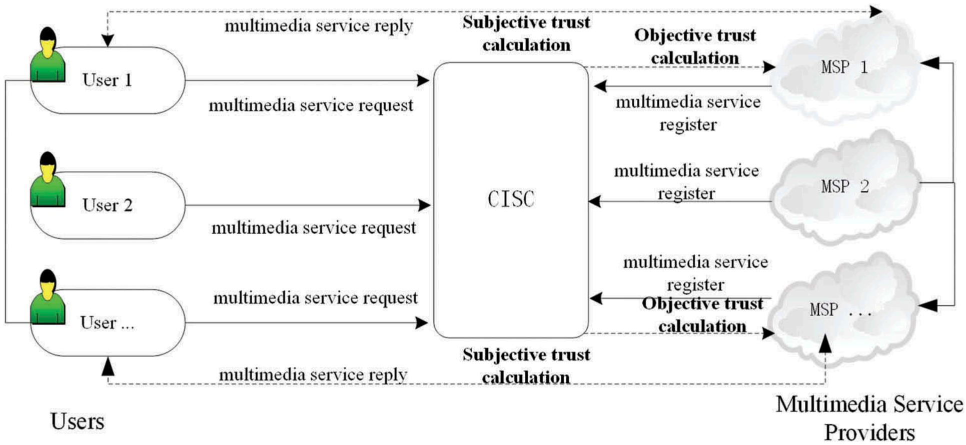


Figure 2. Multimedia services scheduling process in cloud computing environment.

4. Formulation and solution of multimedia services scheduling and provisioning model

In this section, we present the formulation and solution of multimedia services scheduling problem of independent tasks across multi clouds. Key notations are listed in Table 1.

1) Performance Model

There are some independent MSPs in cloud computing environment. Each of them is connected via high-speed network. Let $P = \{p_1, p_2, \ldots, p_m\}$ represent a set of MSPs, available to thousands of users at the same time. Each MSP p_j is characterized by a four tuple, i.e. $p_j = \{id_j, name_j, Tr_j, resList_j\}$. *id*, *name*, *Tr*, *resList* represent the ID, Name, trust degree and resource set of MSPs, respectively. Each resource is presented by $res_j = \{com_j, ep_j, st_j, ti_j, bw_j\}$, com_j, ep_j, st_j, ti_j, bw_j represent the processing capacity, computational price, storage price, traffic price, bandwidth of MSPp_j, respectively.

In this paper, we focus on a class of parallel security-sensitive cloud-based multimedia services, such as video transform, medical image and video render. Each service request consists of a set of

Table 1. The notations.

Notation	Description
P	set of MSPs
p_j	the *j*th MSP
com_j	the processing capacity of MSPp_j
ep_j	computational price of MSPp_j
st_j	storage price of MSPp_j
ti_j	traffic price of MSPp_j
bw_j	bandwidth of MSPp_j
T	set of multimedia tasks
t_i	the *i*th multimedia task
a	the arrival time of a multimedia service request
f	the completion time of a multimedia service request
D	the deadline of executing the multimedia service
M	the cost constraint of executing the multimedia service
$length_i$	the task length of task t_i,
$datasize_i$	the data size of task t_i,
tr_i	the trust constraint of task t_i,
est_j	the remaining execution time of the running task on MSPp_j
ft_{ij}	the finish time of task t_i on MSPp_j

independent tasks. Let a five tuple (T, a, f, D, M)represent such a service request. A set $T = \{t_1, t_2, ..., t_n\}$ represents dynamical independent tasks. Task t_i can be described by a collection of parameters, i.e., $t_i = \{length_i, datasize_i, tr_i\}$, where $length_i$, $datasize_i$ and tr_i are the task length, data size and trust constraint of task t_i, respectively. In this paper, we only focus on inbound traffic, hence $datasize_i$ is the input data of the task. It affects the data transfer time. a and $f$are the arrival time and completion time of an service. D represents the deadline of executing the multimedia service. M represents the cost constraint of executing the multimedia service.

Let est_j represents the remaining execution time of the running task on MSPp_j. It can be determined by the remaining workload size of all tasks that are running on p_j. Let dtt represents the data transmission time. Data transfer occurs when a MSP p_j does not have data of task t_i. The data transmission time dtt is defined as follows:

$$dtt_{ij} = \frac{datasize_i}{bw_j}, \tag{1}$$

let et_{ij} be the execution time of task t_i on MSPp_j, We get

$$et_{ij} = \frac{length_i}{com_j} + dtt_{ij}. \tag{2}$$

Let ft_{ij} be the finish time of task t_i on MSPp_j, then:

$$ft_{ij} = et_{ij} + est_j, \tag{3}$$

Moreover, we use x_{ij}, a binary variable to reflect a mapping task t_i to MSP p_j in multimedia cloud. If x_{ij} is 1, task t_i is allocated to MSPp_j, and is 0 otherwise.

The finish time is also used to determine whether the task's timing constrain can be guaranteed, i.e.

$$x_{ij} = \begin{cases} 0 & if\ ft_{ij} > D, \\ 1\ or\ 0, & if\ ft_{ij} \leq D. \end{cases} \tag{4}$$

Generally, the service charge includes three items: computing, storage and data transfer. Hence, the cost of processing task t_i can be computed as followed:

$$CostF(t_i, p_j) = \frac{l_i}{c_j} \times ep_j + sd_i \times st_j + sd_i \times ti_j. \tag{5}$$

To meet deadline constraint of a service running in the multimedia cloud, we use a picking variable b_j to indicate whether a MSP is used, where $b_j = 1$ means MSP j is used and $b_j = 0$ otherwise. That is, the following must hold:

$$\max_{j=1-m}\left(\left(\sum_{i=1}^{n} et_{ij} \times x_{ij} + est_j\right) \times b_j\right) \leq D. \tag{6}$$

The cost constraint can be defined as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} CostF(t_i, p_j) \times x_{ij} \leq M. \tag{7}$$

2) Trust Model

In general, trust is classified into two categories: subjective trust and objective trust. Subjective trust is related to the users' perception/belief of a MSP based on their own experiences and other relevant knowledge (Li et al. 2014; Fan and Perros 2014). Objective trust is related to reliability, availability, integrity, and other parameters associated with a service (Li et al. 2014, 2015). In this paper, we build a trust model including both subjective trust and objective trust to evaluate the trust of a MSP.

(1) Subjective trust evaluation

To meet the trust constraint of a multimedia service running in a MSP, we apply Bayesian theory to evaluate the subjective trust of MSPs. Bayesian theory is based on subjective probability, which is an opinion and degree of rational belief. It is the probability that someone believes something will happen. Subjective trust relationships usually include direct trust and indirect trust.

The direct trust is achieved by direct transaction history between cloud entities. The transaction history can be described by users' past experience with MSPs.

Assume A and B are two entities in the multi clouds, and their transaction results are expressed by binomial events. Let N denote times trades between them, S is successful trade times; F unsuccessful trade times; and Tr_{dt} denote the probability of successful trade at N + 1 times. Hence,

$$Tr_{dt} = E(Beta(Tr|S+1, F+1)) = \frac{S+1}{S+F+2}, \tag{8}$$

where $0 < Tr_{dt} < 1$, *and* $F, S > 0$.

If cloud entities trade with unfamiliar ones, they cloud adopt or combine indirect trust to help the trust decision. To compute the indirect trust, an entity makes use of its own experience of interactions, but also recommendation information from other entities. Using these recommendations, the user (MSP) computes the number of successful trade and the number of unsuccessful ones for each response.

Let q be the available recommendation number, n_s^m and n_f^m be the successful trade number, the unsuccessful trade number respectively. Those numbers are calculated according to the recommendations responded by an entity. Let n_s^r, n_f^r be the total successful trade number and the unsuccessful trade number in the recommendations, $\hat{\theta}_{rt}$ as the probability of successful trade, we get

$$\hat{\theta}_{rt} = \frac{n_s + \sum_{k=1}^{q} n_s^k + 1}{\sum_{k=1}^{q} n_s^k + \sum_{l=1}^{q} n_f^l + 2}, \tag{9}$$

Entities maybe not collect adequate experience from own history of trade to compute a direct trust. So, we employ a metric called Conf (level of confidence) (Deno and Sun 2008) to aid an entity determine whether the information collected is sufficient for computing the direct trust.

The Conf is achieved as follow:

$$Conf = 1 - \frac{(n_s+1)(n_f+1)}{(n_s+n_f+2)^2(n_s+n_f+3)}. \tag{10}$$

If Conf is low, it indicates that the confidence in the current trust information is not sufficient for computing the direct trust and the entity should collect more information.

Since the trust degree varies by time, we adopt a decay factor to represent the importance of the history information dropping with time. The trade of entities consists of time sequences. Let i represent a certain sequence, s_i and f_i is the successful trade number and the unsuccessful trade number, the formula with the decay factor can be modeled (Wang et al. 2012):

$$s(n) = \sum_{i=1}^{n} s_i \cdot \eta^{(n-i)}, f(n) = \sum_{i=1}^{n} f_i \cdot \eta^{(n-i)}. \tag{11}$$

where $s(n)$, $f(n)$ is the number of successful and failure trade after nth sequence separately, and $0 \leq \eta \leq 1$. When $\eta = 1$, history interactions is abandoned; $\eta = 0$, the latest history record is considered. It should record the whole history interactions and can be solved by applying the following recursive algorithm (Wang et al. 2012).

$$s(i) = s(i-1) \cdot \eta + s_i, f(i) = f(i-1) \cdot \eta + f_i. \tag{12}$$

where $s(i)$ and $f(i)$ are the number of successful and failure trade at the i th sequence. The direct and recommendation trust at time i can be evaluated by formula (8) and (11) with $s(i)$ and $f(i)$ introduced into formula (12).

For two entities A and B in multimedia cloud, the successful trade probability between them is denoted by Tr_{su}. Direct interaction between A and B is denoted by Tr_{dt}. Indirect interaction is represented by Tr_{rt}. Thus, the subjective trust degree can be described as followed:

$$\begin{cases} Tr_{su} = wd \times Tr_{dt} + wr \times Tr_{rt} \\ wd + wr = 1, wd, wr \geq 0 \end{cases}. \tag{13}$$

where wd, wr is the weight of direct trust Tr_{dt} and indirect trust Tr_{rt}.

(2) Objective trust evaluation

From the perspective of objective trust, we mainly focus on reliability, availability, and integrity, three trusted factors of multimedia services.

Reliability is the probability of trusted service for a given duration, and we apply the average task success ratio to indicate this metric. It can be measured by

$$RE = \frac{renum_s}{renum_t} \tag{14}$$

where RE represents reliability; $renum_s$ and $renum_t$ denote the number of successful and total trades that occurred in a time interval, respectively. The average task success ratio is a statistical value for a special MSP. For example, within a given time, if a user asks for twenty services and seventeen are responded and finished. The average task success ratio is 17/20 = 0.85.

Availability is the uptime percentage of multimedia services during a time interval, which can be measured by

$$AV = \frac{avnum_s}{avnum_t} \tag{15}$$

where $0 \leq AV \leq 1$ represents availability; $avnum_s$ and $avnum_t$ denote the uptime and the total time of an trade period, respectively.

Integrity is the data accuracy of the result data, which can be measured by

$$IN = \frac{innum_s}{innum_t} \tag{16}$$

where $innum_s$ is the number of service requests successfully completed by the MSP, $innum_t$ is the number of service requests integrity completed by the MSP.

The objective trust of a MSP can be expressed by

$$Tr_{ob} = w_{RE}RE + w_{AV}AV + w_{IN}IN \tag{17}$$

where w_{RE}, w_{AV}, and w_{IN} denote the weight of RE, AV and IN while evaluating the objective trust of a MSP respectively, $w_{RE} + w_{AV} + w_{IN} = 1$.

(3) Comprehensive trust calculation

According to the above description, the comprehensive trust calculation of a MSP can be given by

$$CT = \alpha Tr_{su} + \beta Tr_{ob} \tag{18}$$

where α and β is the weight of Tr_{su}, Tr_{ob} respectively, $\alpha + \beta = 1$.

(4) Obtain the trust metrics

In general, each MSP must register its multimedia services through the CISC. The user negotiates with the CISC on QoS requirements. To acquire trust metrics, we designed computing agents (CAs), which is a software agent and responsible for collecting and computing objective trust metrics. These metrics can be obtained through calculation and statistics. Such metrics include the average task success ratio, the availability, the integrity of a service etc. To monitor multimedia services, we apply CAs to monitoring the average response time, and the average task success ratio etc. A CA captures data regarding the interaction process between a CISC and a MSP, and responds to requests for monitoring data from CISC. The CA collects monitoring information from the server side continuously, which involves all the performance parameters. The process of obtaining trust metrics is shown in Figure 3.

3) Formalization of optimization problem

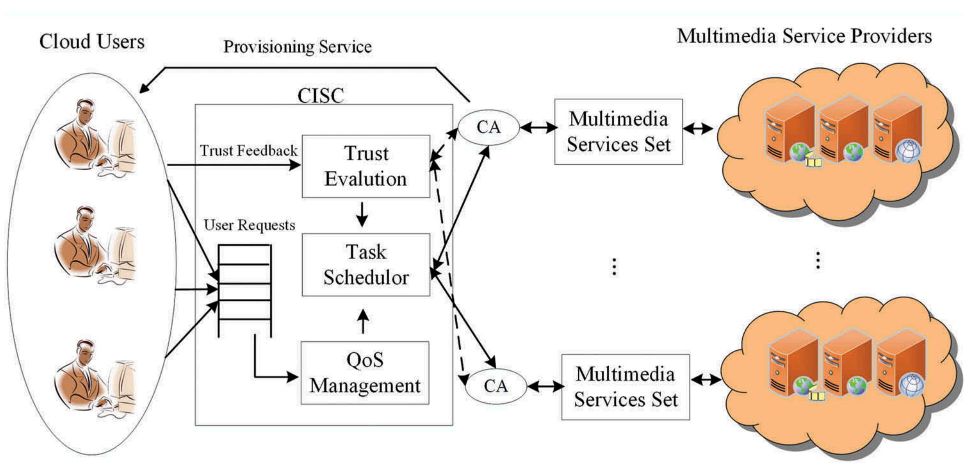


Figure 3. The process of obtaining trust metrics in multimedia cloud computing environment.

Mathematically, a typical model for multimedia applications in the multi clouds can be described as follows:

$$Minimize : Z = \max_{j=1-m}\left(\left(\sum_{i=1}^{n} et_{ij} \times x_{ij} + est_j\right) \times b_j\right), \tag{19}$$

$$s.t.\begin{cases} \max\limits_{j=1-m}\left(\left(\sum\limits_{i=1}^{n} et_{ij} \times x_{ij} + est_j\right) \times b_j\right) \le D \\ \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} CostF(t_i, p_j) \times x_{ij} \le M \\ tr_i \ge MSPTr_j \\ x_{ij}, b_j \in \{0, 1\} \end{cases}. \tag{20}$$

where Z is the object for minimization like time, cost, etc. X_d is the set of feasible solutions satisfying the QoS requirements. The aim of the trust-driven scheduling is to map every t_i onto some p_j in order to achieve minimum execution time while meeting the constraints in terms of budget and trust.

Because many of cloud-based multimedia services are not risk-free during the execution, it is essential for these MSPs to have security awareness. Therefore, we derive the risk probability to quantitatively analyze the risk rate for a multimedia service scheduled on MSPs. We suppose the risk rate is a function of trust levels. The distribution of the risk for any fixed time interval follows a Poisson probability distribution. The risk rate model is used for illustration purpose only. Thus, the service's risk rate on a specific MSP can be presented as follows:

$$pr(t_i, p_j) = \begin{cases} 0, & if\ t_i.Tr \le p_j.trust \\ 1 - e^{-(t_i.Tr - p_j.trust)}, & if\ t_i.Tr > p_j.trust \end{cases}. \tag{21}$$

To find the optimal solution for a given problem that schedules n tasks on m MSPs with several restrictions, some heuristics must be employed to solve this problem in a real multimedia cloud environment since the number of MSPs and services is very large.

5. Multimedia service scheduling and provisioning algorithm in cloud computing environment

This section presents the proposed trust multimedia scheduling algorithm (TMSS) that allocates multimedia cloud resources to meet the demand in term of cost and trust of the submitted multimedia tasks.

The proposed TMSS algorithm is outlined as followed: In the pseudo code of TMSS optimization, a multimedia service enters the waiting queue according to its deadline, its tasks are sorted by their size (see line 1–2). Next, the trust degree of cloud entities is calculated (see line 3–9). Then, it adds all the available MSPs in the set *trustP* (see lines 10–13). The execution cost and time of each available MSP is calculated (see lines 16–17). After that, the tasks in multimedia service can be scheduled to fit MSP to meet the demand of cost and deadline (see lines 20–23).

Algorithm TMSS

1. Sort media service according to the deadline, select the earliest deadline one in the arrival queue;
2. initialize the cost overhead of service, TotalCost = 0;
3. **for** MSP p_j of MSPs **do**
4. calculating the direct trust degree of MSPs using Equation (8)
5. calculating the indirect trust degree of MSPs using Equation (9)
6. calculating the subjective trust degree of MSPs using Equation (13)
7. calculating the objective trust degree of MSPs using Equation (17)
8. calculating the comprehensive trust degree of MSPs using Equation (18)
9. **end for**
10. **for** task t_i of the service selected **do**
11. **for** each MSP p_k of multimedia cloud
12. computing $pr(t_i, P_k)$ using Equation (20);
13. **end for**
14. search the trust MSP set P′ with $pr(t_i, P_j) < \theta, \theta$ is the risk threshold;
15. **While** (trustP not NULL) **do**
14. **If** $f_i \leq D$ and TotalCost<M **then**
15. **for** each MSP $p_j \in trustP$ **do**
16. calculate and save the cost c_{ij} for executing t_i on MSP p_j
17. calculate and save et_{ij}
18. **end for**
19. find the p_{best} that has the shortest completion time
20. **if** p_{best} exists
21. assign task t_i to pr_{best}
22. add the cost of running t_i on p_{best} to TotalCost
23. **end if**
24. **end if**
25. **end while**
26. **end for**

6. Performance evaluation

In this section, we first illustrate the simulated experiment environment, and then introduce the performance metrics. Next, we introduce the related algorithms. Finally, we detail the experiment results.

6.1. *Simulated experiment environment*

In this paper, we assume the multimedia services are provisioned through the type of virtual machines. Using the Java programming tools, we implemented a simulation environment, which is a cross-clouds platform including 10 MSPs, 100 virtual resources and some of users (SRs). Two assumptions are made: (1) a SR can trade with other SRs and offer the recommendations; (2) a SR can trade with the same type of MSPs.

The simulations run on a machine with an Intel Core 2 Duo CPU and 4 GB RAM. Each incoming task is set to have 200/400 MB input data size, 10^6–10^7 Million Instructions and 0–1.0 trust requirement. The processor speed follows a uniform distribution in the interval [800, 1000] MIPS. The pricing model of resources based on Amazon prices, is summarized in Table 2. The failure rate follows a uniform distribution in the interval [0, 0.050].

Table 2. Resource pricing model.

Resource	Price($)
Computation	1–2.00(per hour)
Network(In)	0.001(per GB)
Storage	0.001 (per GB per hour)

6.2. *Performance metrics*

To evaluate the trust integration performance, we use the following metrics:

Completion time: the duration time needed for the system to complete the execution of all tasks belonging to the same service.

Cost: the total cost in monetary units accumulated for an entire service execution.

Success rate: it is calculated for an service execution as a ratio between the number of successfully executed tasks and the number of the total scheduled tasks. MSPs that are error prone or those who fail, lead to smaller execution efficiency since the tasks they received must be rescheduled.

6.3. *Related algorithms*

To demonstrate the merit of TMSS, we compare it with three famous algorithms, i.e., MCT, Greedy, and TWFS(Tan et al. 2014). MCT: sorts MSPs by their computational capacity and selects the MSPs with the minimum completion time for the corresponding task. On the contrary, the Greedy approach sorts MSPs by their cost and selects the MSP with the minimum cost. TWFS achieve minimum execution cost and time while meeting the constraints of trust. The key idea of the TWFS algorithm is to find the optimum solution with the deadline constraint by adjusting the weights of time and cost.

6.4. *Simulation experiments*

6.4.1. *Effects of task number*

The experiment is to test the MCT, TWFS, Greedy, TMSS algorithms with different task number. The number of tasks varies from 100 to 500 with an increment of 100. This time we set the every job deadline to be a reasonable integer. The results of execution time are figured in Figure 4. Figure 4 show that the TMSS can averagely perform 89.3 % less time than the MCT in terms of the completion time. The completion time of TMSS is shortest among the four algorithms. Obviously, the TMSS performs significantly better than other algorithms in latency comparisons.

The experiment result shows that the proposed scheduling integrating trust me chiasm can shorten the completion time of services compared to the MCT and Greedy algorithm, which do not load trust mechanisms. MCT shorten the services execution time by assigning them to the MSPs

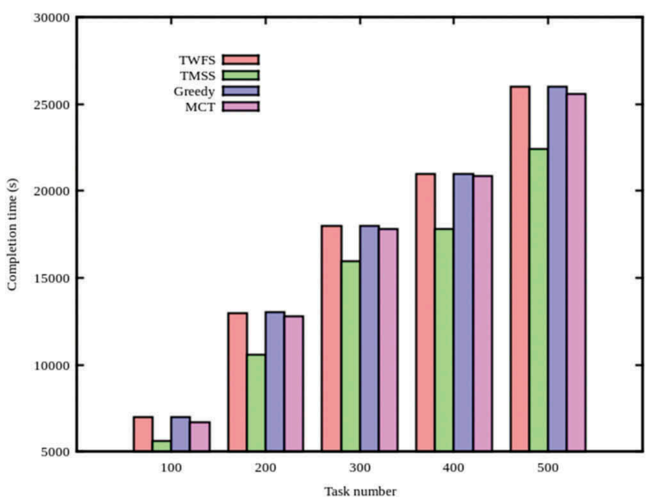


Figure 4. Comparison of completion time varying task number.

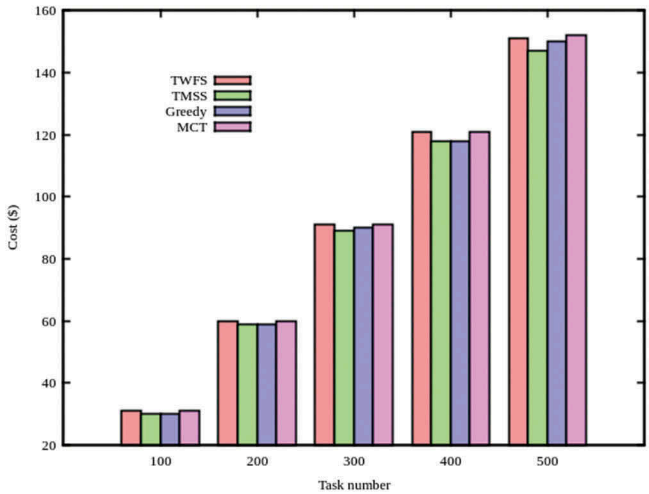


Figure 5. Comparison of cost varying task number.

which has the shortest completion time. But they ignore resources integrity and sensitive tasks may be sent to untruthful MSPs. Hence the execution is delayed and necessary to be re-scheduled. The services completion time is postponed. While our schema considers the trust degree of the candidate MSPs in the scheduling process and reduces the probability of failure scheduling. The whole service execution time is shortened. With task number rising, trust-driven scheduling schema describes more advantage.

Figure 5 shows the result of the cost for executing jobs of 100, 200, 300,400 and 500 independent tasks by TMSS, MCT,T WFS and Greedy. From Figure 5, we can judge that the execution cost of TMSS algorithm is close to that of Greedy's which is the cheapest. It indicates that the TMSS algorithm outperforms other two algorithms. In this experiment, the TMSS spends about 20.7 % less money than the MCT in average. While the cost of MCT, Greedy and TWFS is close to each other. The reason is that the TMSS considers the trust constraint as the most important constraint, while the MCT focuses on lowering the execution time and spends more money.

6.4.2. *Effect of varying deadline*

In this experiment, we investigate the effect of deadline on the performance of the trustful scheduling algorithm. Deadline is computed according to the minimum execution time produced by MCT. We change deadline from 10000 seconds to 14000 seconds for 200 tasks. The deadline for 500 tasks range from 26000 seconds to 30000 seconds. Figure 6 depicts the completion time result of our experiment. In Figure 6(a), it is observed that for task number 200, an increase in deadline also gives rise to the completion time. We can see that the completion of TMSS is constant while deadline is large than 11000 seconds. This is TMSS takes the shortest time to execute tasks. From Figure 6(b), the same takes place.

Figure 7(a) shows the results of scheduling several jobs consisting of 200 tasks totally. Given a very strict deadline such as 10000 seconds, all schedulers present a similar, low satisfaction rate because of the limitation of available resources. With a loose deadline such as 12000 seconds, the TMSS can achieve better success ratio than the other three schedulers. Figure 7(b) shows that, for scheduling several jobs consisting of totally 500 tasks, the TMSS can reach 100 % success ratio for a deadline of 27000 seconds while the others cannot even reach 87 % success ratio for a deadline of 27000 seconds. The major reason why the TMSS is better in

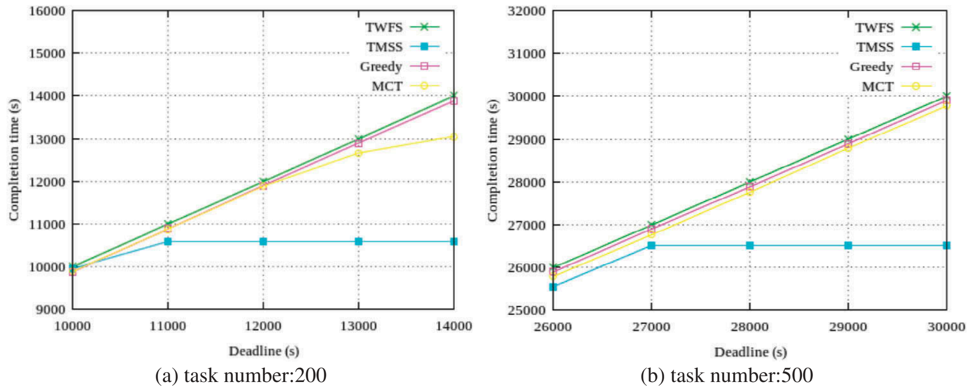


Figure 6. Comparison of completion time varying deadline.

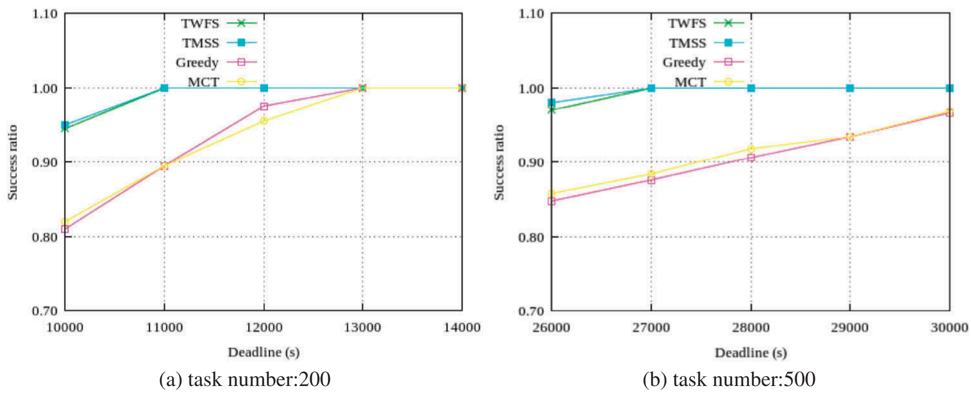


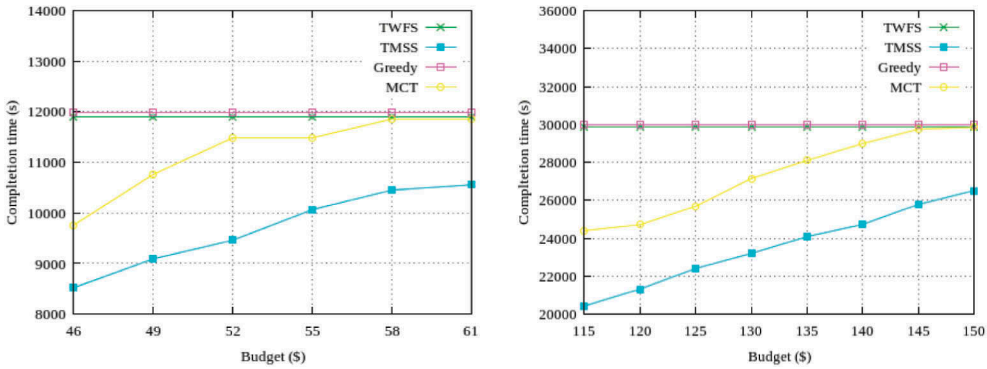
Figure 7. Comparison of success ratio varying deadline.

success ratio is that the TMSS tries to send tasks to trust MSP and assure the normal execution of tasks.

6.4.3. *Effect of varying budget*

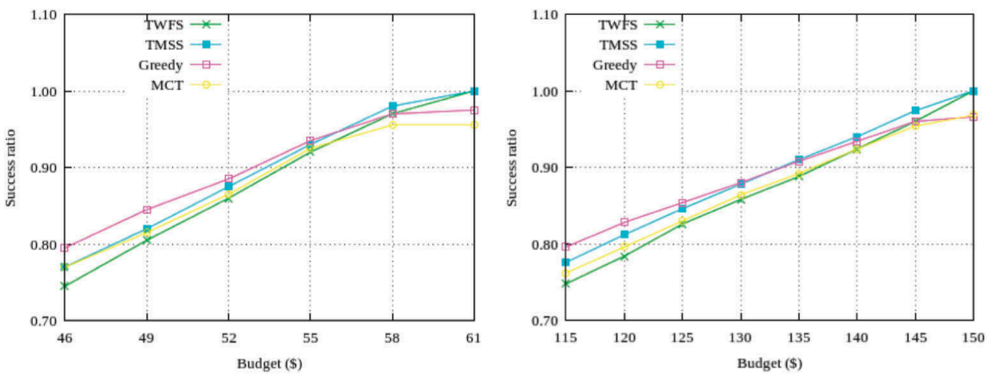
In this experiment, the budget is computed according to the minimized cost produced by Greedy. For 200 tasks, the budget varies from 46$ to 61$. The result of completion time is presented by Figure 8(a). For 500 tasks, the budget ranges from 115$ to 150$. The result of completion time is presented by Figure 8(b). From Figure 8(a,b), we can see that the completion time of tasks produced by TWFS is closer to Greedy. This is because both scheduling algorithms try to schedule tasks to cheapest or maximized utility resource. The completion time produced by TMSS is shortest among the four algorithms. TMSS spends less than 20 percent than that of MCT. The experiment result shows that the new scheduling model including trust requirement can shorten the services completion time while compared to the MCT and Greedy which do not provide trust mechanisms.

From Figure 9, we can see that: in multimedia cloud environment with some untruthful MSPs, scheduling model including trust mechanism can assure higher success rate because it efficiently eliminates untruthful MSPs according to passed trade information and other trust users' recommendation. The ratio of successful execution of TMSS is much higher than TWFS's and MCT's. This demonstrates that the trustful scheduling algorithm can guarantee the successful execution of tasks.



(a) task number:200 (b) task number:500

Figure 8. Comparison of completion time varying budget.



(a) task number:200 (b) task number:500

Figure 9. Comparison of success ratio varying budget.

7. Conclusion and future work

In this paper, we develop and evaluate a novel scheduling and provision scheme for multimedia services in cloud computing environment. Trust mechanism can dynamically assess the trust degree of multimedia service provider by Bayesian method and integrated the trust metric into scheduling, which ensures the execution success rate of security-critical multimedia tasks. The proposed trustful scheduling scheme can heuristically seek approximate optimal resource allocations and meet the requirement of trust and deadline for multimedia services. The performance of TMSS algorithm is compared with three well-known scheduling algorithms for multimedia service: MCT, TWFS and Greedy algorithms. Detailed experiment results demonstrate the feasibility and efficiency of the proposed scheme.

As part of our future work, we plan to extend the TMSS algorithm for dependent multimedia services in cloud computing environments.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their helpful comments and suggestions. The work was supported by the National Natural Science Foundation (NSF) under grants (No.61472294, No. 61672397), Applied Basic Research Project of WuHan (No.2015010101010021), Supported by Key Laboratory of Spatial Data Mining & Information Sharing of Ministry of Education, Fuzhou University□No. 2016LSDMIS05□, Program for the High-end

Talents of Hubei Province. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the the National Natural Science Foundation (NSF); [No.61472294, No. 61672397]; Applied Basic Research Project of WuHan; [No.2015010101010021]; Key Laboratory of Spatial Data Mining & Information Sharing of Ministry of Education, Fuzhou University; [No. 2016LSDMIS05].

References

An, N.T., C.-T. Huynh, B. Lee, C.S. Hong, and E.-N. Huh. 2014. “An Efficient Block Classification for Media Healthcare Service in Mobile Cloud Computing.” *Multimedia Tools and Applications* 74 (14): 5209–5223.

Deno, M. K., and T. Sun. 2008. “Probabilistic Trust Management in Pervasive Computing.” In *Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, Shanghai, China, 610–615. IEEE Computer Society Press.

Dong, G., and J. Dai. 2008. “Scheduling Algorithm for Multimedia Services in IEEE 802.16e OFDM System.” *Transactions of Tianjin University* 14 (1): 31–36. doi:10.1007/s12209-008-0007-9.

Fan, W., and H. Perros. 2014. “A Novel Trust Management Framework for Multi-Cloud Environments Based on Trust Service Providers.” *Knowledge-Based Systems* 70: 392–406. doi:10.1016/j.knosys.2014.07.018.

Fard, H. M., R. Prodan, and T. Fahringer. 2013. “A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments.” *IEEE Transactions on Parallel and Distributed Systems* 24 (6): 1203–1212. doi:10.1109/TPDS.2012.257.

Javadi, B., J. Abawajy, and R. Buyya. 2012. “Failure-Aware Resource Provisioning for Hybrid Cloud Infrastructure.” *Journal of Parallel and Distributed Computing* 72 (10): 1318–1331. doi:10.1016/j.jpdc.2012.06.012.

Chen, J.-C., K.M. Sivalingam, P. Agrawal, and R. Acharya. 1998. “On Scheduling of Multimedia Services in a Low-Power MAC for Wireless ATM Networks.” In *Proceedings of Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium*, Boston, MA, 243–247, Vol. 1.

Kamvar, S. D., M. T. Schlosser, and H. Garcia-Molina. 2003. “The Eigentrust Algorithm for Reputation Management in P2p Networks.” In *Proceedings of Proceedings of the 12th international conference on World Wide Web*, Budapest, Hungary, 640–651. ACM.

Li, W. J., J. Wu, Q. Zhang, K. Hu, and J. Li. 2014. “Trust-Driven and Qos Demand Clustering Analysis Based Cloud Workflow Scheduling Strategies.” *Cluster Computing* 17 (3): 1013–1030. doi:10.1007/s10586-013-0340-1.

Li, X., H. Ma, F. Zhou, and X. Gui. 2015. “Service Operator-Aware Trust Scheme for Resource Matchmaking across Multiple Clouds.” *IEEE Transactions on Parallel and Distributed Systems* 26 (5): 1419–1429. doi:10.1109/TPDS.2014.2321750.

Li, Y., L. Zhuo, and H. Shen. 2013. “An Efficient Resource Allocation Method for Multimedia Cloud Computing.” In *Proceedings of 4th International Conference on Intelligence Science and Big Data Engineering*, Beijing, China, 246–254. Springer-Verlag Berlin Heidelberg.

Lin, C. C., H. H. Chin, and D. J. Deng. 2014. “Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System.” *IEEE Systems Journal* 8 (1): 225–234. doi:10.1109/JSYST.2013.2256320.

Lin, S., X. Zhang, Q. Y, H. Qi, and S. Ma. 2013. “Parallelizing Video Transcoding with Load Balancing on Cloud Computing.” In *Proceedings of Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE*, Beijing, China, 2864–2867.

Lu, P., Q. Sun, K. Wu, and Z. Zhu. 2015. “Distributed Online Hybrid Cloud Management for Profit-Driven Multimedia Cloud Computing.” *IEEE Transactions on Multimedia* 17 (8): 1297–1308. doi:10.1109/TMM.2015.2441004.

Lu, W., A. L. Varna, and M. Wu. 2010. “Security Analysis for Privacy Preserving Search of Multimedia.” In *Proceedings of 2010 17th IEEE International Conference on Image Processing (ICIP)*, Hong Kong, 2093–2096. IEEE Computer Society Press.

Nan, X., H. Yifeng, and G. Ling. 2011. “Optimal Resource Allocation for Multimedia Cloud Based on Queuing Model.” In *Proceedings of 2011 IEEE 13th International Workshop on Multimedia Signal Processing*, Hangzhou, China, 1–6.

Qu, C., and R. Buyya. 2014. “A Cloud Trust Evaluation System Using Hierarchical Fuzzy Inference System for Service Selection.” In *Proceedings of 2014 IEEE 28th International Conference on Advanced Information Networking and Applications (AINA)*, Victoria, British Columbia, Canada, 850–857. IEEE Computer Society.

Ren, S., and M. van der Schaar. 2013. "Efficient Resource Provisioning and Rate Selection for Stream Mining in a Community Cloud." *IEEE Transactions on Multimedia* 15 (4): 723–734. doi:10.1109/TMM.2013.2240673.
Shini, S. G., T. Thomas, and K. Chithraranjan. 2012. "Cloud Based Medical Image Exchange-Security Challenges." *Procedia Engineering* 38: 3454–3461. doi:10.1016/j.proeng.2012.06.399.
Song, B., M. M. Hassan, A. Alamri, A. Alelaiwi, Y. Tian, M. Pathan, and A. Almogren. 2016. "A Two-Stage Approach for Task and Resource Management in Multimedia Cloud Environment." *Computing* 98 (1–2): 119–145. doi:10.1007/s00607-014-0411-z.
Tan, M., and X. Su. 2011. "Media Cloud: When Media Revolution Meets Rise of Cloud Computing." In *Proceedings of 6th IEEE International Symposium on Service Oriented System Engineering*, Irvine, CA, 251–261. IEEE Computer Society Press.
Tan, W., Y. Sun, L. X. Li, G. Lu, and T. Wang. 2014. "A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing." *IEEE Systems Journal* 8 (3): 868–878. doi:10.1109/JSYST.2013.2260072.
Wang, C., L. Xing, H. Wang, Y. Dai, and Z. Zhang. 2014a. "Performance Analysis of Media Cloud-Based Multimedia Systems with Retrying Fault-Tolerance Technique." *IEEE Systems Journal* 8 (1): 313–321. doi:10.1109/JSYST.2013.2253042.
Wang, H., S. Wu, M. Chen, and W. Wang. 2014b. "Security Protection between Users and the Mobile Media Cloud." *IEEE Communications Magazine* 52 (3): 73–79. doi:10.1109/MCOM.2014.6766088.
Wang, S., Z. Liu, Q. Sun, H. Zou, and F. Yang. 2014c. "Towards an Accurate Evaluation of Quality of Cloud Service in Service-Oriented Cloud Computing." *Journal of Intelligent Manufacturing* 25 (2): 283–291. doi:10.1007/s10845-012-0661-6.
Wang, S., A. Zhou, C. H. Hsu, X. Xiao, and F. Yang. 2016. "Provision of Data-Intensive Services through Energy- and Qos-Aware Virtual Machine Placement in National Cloud Data Centers." *IEEE Transactions on Emerging Topics in Computing* 4(2): 290-300.
Wang, W., G. Zeng, D. Tang, and J. Yao. 2012. "Cloud-DLS: Dynamic Trusted Scheduling for Cloud Computing." *Expert Systems with Applications* 39 (3): 2321–2329. doi:10.1016/j.eswa.2011.08.048.
Wen, H, Z. Hai-ying, L. Chuang, and Y. Yang. 2011. "Effective Load Balancing for Cloud-Based Multimedia System." In *Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology*, Dalian, China, 165–168. IEEE Computer Society Press.
Wu, C. H., W. H. Ip, C. Chan, Y. Zheng, and Z. M. Lu. 2010. "A Case Study on Multimedia E-Tailing Services: Can Watermarks Help Build Customer Relationships?" *International Journal of Services Technology and Management* 14 (1): 58–76. doi:10.1504/IJSTM.2010.032881.
Xiaoyong, T., K. Li, Z. Zeng, and B. Veeravalli. 2011. "A Novel Security-Driven Scheduling Algorithm for Precedence-Constrained Tasks in Heterogeneous Distributed Systems." *IEEE Transactions on Computers* 60 (7): 1017–1029. doi:10.1109/TC.2010.117.
Yang, C., W. Shih, L. Chen, C.-T. Kuo, F.C. Jiang, and F.Y. Leu. 2015. "Accessing Medical Image File with Co-Allocation HDFS in Cloud." *Future Generation Computer Systems* 43-44: 61–73. doi:10.1016/j.future.2014.08.008.
Zhou, L., A. V. Vasilakos, N. Xiong, Y. Zhang, and S. Lian. 2011. "Scheduling Security-Critical Multimedia Applications in Heterogeneous Networks." *Computer Communications* 34 (3): 429–435. doi:10.1016/j.comcom.2010.01.009.
Zhu, W., C. Luo, J. Wang, and S. Li. 2011. "Multimedia Cloud Computing." *IEEE Signal Processing Magazine* 28 (3): 59–69. doi:10.1109/MSP.2011.940269.
Zhu, X., and P. Lu. 2009. "A Two-Phase Scheduling Strategy for Real-Time Applications with Security Requirements on Heterogeneous Clusters." *Computers & Electrical Engineering* 35 (6): 980–993. doi:10.1016/j.compeleceng.2008.11.022.